Network-Guided Covariate Selection and Downstream Applications

Wanjie Wang

National University of Singapore

BIRS-IASM Workshop: Harnessing the power of latent structure models and modern Big Data learning

December 11-15, 2023

High Dimensional Data

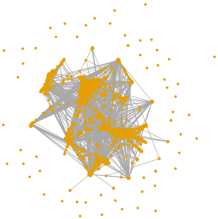
- Data matrix $X \in R^{n \times p}$: n samples, p covariates
- High dimensional data
  - large p
  - increasing with n in the asymptotic setting
  - challenge: curse of dimension
- Problems of interest:
  - **variable selection**
  - testing, clustering, regression, etc.
- Tons of literature in past decades

Example: Complex Data Example

LastFM app user data (*Rozemberczki and Sarkar (2020)*):

```
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0
```

List of their liked artists



For each user:

- User-specific data: the list of artists each user likes
- sample size: hundreds; number of artists: 2201
- **New data source:** the mutual friendship between users

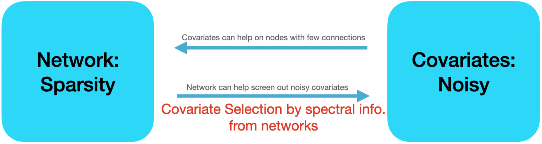
Covariates with Network Data

In this example, we observe the following data format:

$$(Covariates\ X \in \mathcal{R}^{n\times p}, Network\ A \in \mathcal{R}^{n\times n})$$

- X: high dimensional data suffer the curse of dimensionality
- A: difficult to observe for new data; sparsity
- Applications: psychological study; social platform; brain image data; etc.
- Idea: combine them

Covariates with Network Data



Network: Sparsity

Covariates can help on nodes with few connections

Network can help screen out noisy covariates

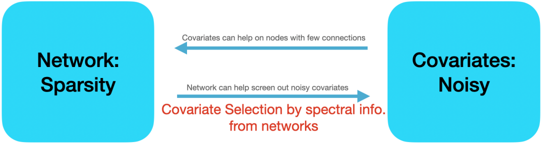
Covariate Selection by spectral info. from networks

Covariates: Noisy

For high-dimensional covariates:

- Variable selection?
  - Use the network to select useful covariates for regression and clustering (Gu and Han (2011), Samain et al. (2018), Singh et al. (2020), Wang and Chen (2021), Zhao et al. (2022), Zhu et al. (2019) and more)
- Next step regression and clustering?
  - Data (A, X, X_{new}). Results on X_{new}

For this talk


Network:
Sparsity
Covariates can help on nodes with few connections
Network can help screen out noisy covariates
Covariate Selection by spectral info. from networks
Covariates:
Noisy

- Variable selection
  - We propose a rate-optimal **network-guided covariate selection** algorithm
  - A: spectral method; robust to models
- Consider (A, X, X_{new})
  - On the selected covariates, we then propose algorithms for **clustering and regression**

Part I: Algorithm: Network-Guided Covariate Selection

High Dimensional Covariates Selection

$$X = [X_1, X_2, \cdots, X_p], \qquad X_j \in \mathcal{R}^n$$

Covariate Selection:

Step 1 Propose a statistic $t_j = t(X_j)$ and find the p-value of t_j

Step 2 Use the p-values to select covariates

Latent Structure in the Covariates

Recall: n samples, p covariates

- Data matrix: $X \in \mathcal{R}^{n \times p}$, $A \in \mathcal{R}^{n \times n}$
- Latent information: $Y \in \mathcal{R}^{n \times K}$ (unknown)

Latent Structure in the Covariates

Recall: n samples, p covariates

- Data matrix: $X \in \mathcal{R}^{n \times p}$, $A \in \mathcal{R}^{n \times n}$
- Latent information: $Y \in \mathcal{R}^{n \times K}$ (unknown)
- For j-th covariate:

$$E[X_j] = YM_j, \text{where } M_j \in \mathcal{R}^K$$

- Large amount of covariates are *useless*:

$$M_j = 0, \text{ for most } j \text{ in } 1, 2, \cdots, p$$

Goal: Identify the set $S = \{j : \|M_j\| \neq 0\}$

Test Statistic t_j

Simplification: $X_j \sim N(Y_{n\times K}(M_j)_{K\times 1}, I_n)$

- **Only X** is available, then the optimal stat is

$$X_j^T X_j \sim \chi^2_n(\|YM_j\|^2)$$

Unsupervised

- **Latent info. Y** is known, let $Y = \Xi_{n\times K}\Lambda U_{K\times K}$,

$$X_j^T \Xi\Xi^T X_j \sim \chi^2_K(\|\Xi^T YM_j\|^2) = \chi^2_K(\|\Lambda UM_j\|^2)$$

Supervised

- **A** and **X** are known
  - Using A to guess Ξ, not Y

Latent Structure in the Network

- The latent info. Y plays an important role in $E[A]$
  - Eg1: SBM and its generalizations: $Y_i \in \{0, 1\}^K$,

$$E[A] = YBY^T$$

  - Eg2: Degree-corrected SBM: $E[A] = \Theta YBY^T \Theta$
  - Eg3: Random Dot Product Graph: $Y_i \in$ Unit ball in $\mathcal{R}^K$,

$$E[A] = \rho_n YY^T$$

- Exploring A will provide info. of Y
- The effects of Y depends on the model

Spectral Methods on the Network

Spectral method is generally applied in network analysis:

- Community Detection:
    - Find the top eigenvectors of the adjacency matrix A or the Laplacian matrix $L = D^{-1/2}AD^{-1/2}$
    - Normalize the eigenvectors (many variants)
    - Apply k-means to the normalized eigenvectors
    - *Rohe, Chatterjee, and Yu (2011), Jin (2015), Zhang, Levina and Zhu (2020), etc.*
- Latent position embedding
    - Top eigenvectors of A or the Laplacian L
    - Latent positions: eigenvector*$\sqrt{\text{eigenvalue}}$
    - *Lyzinski et al. (2015), Rubin-Delanchy et al. (2021), Priebe et al. (2018)*
- Connected to the modularity approach
    - *Newman (2006), Newman (2013)*

Spectral Methods in Network

- Spectral methods are useful to extract the info. about Y
  - works for various models, no matter how Y works
- For an accurate estimation of Y, spectral method may suffer
  - column-wise constant factor: $\sqrt{\text{eigenvalue}}$
  - row-wise constant factor
  - choice of the number of eigenvectors, i.e., K
- We only need Ξ, not the exact Y
  - Our goal is to reduce d.f. from n to $O(1)$, so $\hat{K} \geq K$ also works
  - column-wise constant factor doesn't matter
  - row-wise constant factor can be controlled

Step 1: Network-Guided Test Statistic

Input: (A, X), tuning parameter $\hat{K}$: a constant inflation of K

- Find $\hat{\Xi}$:
  - Let ξ_i be the i-th top eigenvector of A
  - $\hat{\Xi} = [\xi_1, \cdots, \xi_{\hat{K}}]$
- Use $\hat{\Xi}$ to find the test statistic
  - Let the test statistic

$$t_j = \|\hat{\Xi}^T X_j\|^2 = \sum_{k=1}^{\hat{K}} (\xi_k^T X_j)^2$$

Step 1: Network-Guided Test Statistic

Input: (A, X), tuning parameter $\hat{K}$: a constant inflation of K

- Find $\hat{\Xi}$:
  - Let ξ_i be the i-th top eigenvector of A
  - $\hat{\Xi} = [\xi_1, \cdots, \xi_{\hat{K}}]$
- Use $\hat{\Xi}$ to find the test statistic
  - Let the test statistic

$$t_j = \|\hat{\Xi}^T X_j\|^2 = \sum_{k=1}^{\hat{K}} (\xi_k^T X_j)^2$$

- Find the p-value

$$\pi_j = P(\chi^2_{\hat{K}} \geq t_j)$$

Step 2: Threshold

- Smaller p-value means larger prob. to be in S
- A cut-off value of p-values
- We use the Higher Criticism Thresholding

Higher Criticism Thresholding (HCT)

The idea goes back to John Tukey:

1. *(Find p-value)* Find the p-value of each covariate, say π_i
2. *(Ordering)* Order them as $\pi_{(1)} \leq \pi_{(2)} \leq \cdots \leq \pi_{(p)}$
3. *(Decide the cut-off)* Calculate the Higher Criticism score

$$HC(j) = \sqrt{p}\frac{j/p - \pi_{(j)}}{\sqrt{\pi_{(j)}(1 - \pi_{(j)})}}$$

4. Let $\hat{s} = \max_j HC(j)$
5. The selected covariates are $\hat{S} = \{j : \pi_j \leq \pi_{(\hat{j})}\}$

Algorithm: Network-Guided Covariate Selection

Input: Network A, covariates X, tuning parameter $\hat{K}$

Step 1 [1]: Test statistic and p-value

1. Find the top $\hat{K}$ eigenvectors of A, denoted as $\xi_1, \cdots, \xi_{\hat{K}}$
2. Define the test stat $t_j = \sum_{k=1}^{\hat{K}} (\xi_k^T x_j)^2$
3. Find p-values that $\pi_j = P(\chi^2_{\hat{K}} > t_j)$

Step 2 : Higher Criticism Thresholding

1. Order the p-values so that $\pi_{(1)} \leq \cdots \leq \pi_{(p)}$. Define

$$HC(j) = \sqrt{p}(j/p - \pi_{(j)})/(\sqrt{\pi_{(j)}(1 - \pi_{(j)})})$$

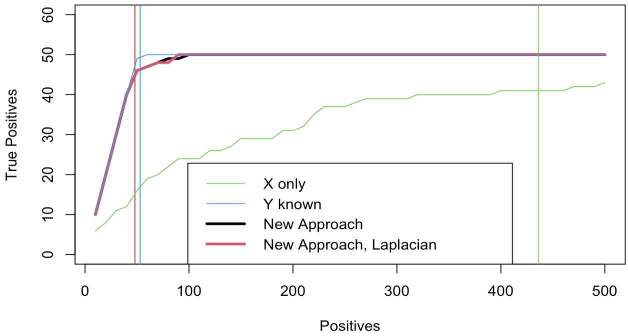
2. The maximizer $\hat{j} = \arg\max_{1 \leq j \leq p/2} HC(j)$
3. $\hat{S} = \{j : \pi_j \leq \pi_{(\hat{j})}\}$

Output: The set of selected covariates $\hat{S}$

[1] Here we can also use the Laplacian L

Network-Guided test is powerful

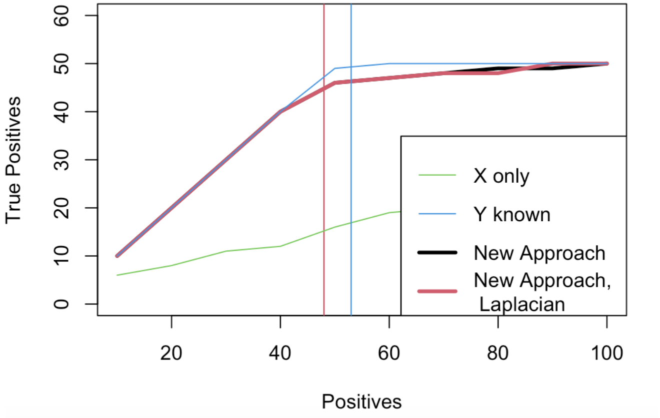
Set $n = 600$, $p = 1200$, $K = 3$, $S = 50$ useful features.
For $j \in \mathcal{S}$, the mean vector $M_j \sim N(0.25, (0.05)^2 I_K)$



- The network largely improves the test power
- Data-driven HCT makes a good selection

Network-Guided test is powerful: Zoom-in

We zoom-in the left panel of that figure



- New test statistic almost achieves the optimal selection

Part II: Optimality when the signals are rare and weak

Notations

- n samples; p covariates
- Network A: $n \times n$ matrix, symmetric, $A_{ij} \in \{0, 1\}$
- Covariates X: $n \times p$ matrix

$$X_j \sim N(YM_j, I_n)$$

- Sparsity:

$$|\mathcal{S}| = |\{j : \|M_j\| > 0\}| = p^{1-\beta}, \qquad \beta > 0$$

- Weakness:

$$\|M_j\| \to 0.$$

Consistency

- Define the signal strength

$$\kappa_A = \min_{j \in \mathcal{S}} \|M_j^T Y^T \Xi^{(\hat{K})}\|^2.$$

Theorem (Consistency)

Suppose the assumptions hold. If $\kappa_A \geq \max\{16(1-\beta), 14\} \log p$*, then with a high prob.,*

$$\max_{i \in \mathcal{S}} \pi_i < \min_{i \notin \mathcal{S}} \pi_i.$$

Furthermore, the network-guided covariate selection algorithm satisfies that

$$\mathcal{S} \subset \hat{S}, \qquad |\hat{S} \backslash \mathcal{S}| \leq C \log^2 p \ll |\mathcal{S}|.$$

Remarks

In the theorem, the signal strength is summarized by

$$\kappa_A = \min_{j \in \mathcal{S}} \|M_j^T Y^T \Xi^{(\hat{K})}\|^2.$$

- When $\Xi^{(\hat{K})} = Y$, then $\kappa_A = \min_{j \in \mathcal{S}} \|(Y^T Y) M_j\|^2$.
- It means, when $Y^T Y \sim nI$, we need $\|M_j\|^2 \geq C \log p/n$
- Of course, $\Xi^{(\hat{K})} \neq Y$, then what the condition $\kappa_A \geq C \log p$ means?
  - Random dot product graph
  - Degree-corrected Stochastic Blockmodel

Random Dot Product Graph

Consider the a special case in the latent position model

$$A_{i,j} \sim Bernoulli(\rho_n Y_i^T Y_j), \qquad Y_i \overset{i.i.d.}{\sim} F$$

- Y_i: the latent position of sample i
- ρ_n: the network density parameter

- The domain of F is a subset in the unit ball in $\mathcal{R}^K$ and $Y_i^T Y_j \geq 0$.
- $Cov(Y_i) \in \mathcal{R}^{K\times K}$ has a full rank
- For any realization Y_i, $Y_i^T E[Y_j] \geq c > 0$
- $n\rho_n \geq c_d \log n$ for a constant $c_d > 0$.

Consistency under RDPG

Corollary (Consistency under RDPG)

Under RDPG with $n\rho_n \geq c_d \log n$. Let $K \leq \hat{K} = O(1)$, then there is a constant c, so that

$$\kappa_A \geq cn \min_{j \in \mathcal{S}} \|M_j\|^2.$$

Therefore, $\mathcal{S}$ can be almost exactly recovered when

$$\min_{j \in \mathcal{S}} \|M_j\| \geq c\sqrt{\log p/n}.$$

- The minimum signal strength is $\|M_j\| = O(\sqrt{\log p/n})$

Degree-Corrected SBM

Degree-Corrected SBM:

$$A_{i,j} \sim \mathit{Bernoulli}(\theta_i \theta_j Y_i^T B Y_j), \qquad Y_i \in \{0,1\}^K.$$

- Y_i is the community membership vector
- $B \in \mathcal{R}^{K \times K}$ is the community by community matrix
- θ_i denotes the heterogeneity among samples

- B has a rank of K
- $n_k/n \geq c > 0$ for each community k
- there is $C > 0$, so that $C\theta_i \geq \max_i \theta_i$ for $i \in [n]$

Consistency under DCSBM

Corollary (Consistency under DCSBM)

Consider DCSBM where $n \max_i \theta_i^2 \geq C \log n$ *for a constant* $C > 0$. *Let* $K \leq \hat{K} = O(1)$, *then there is a constant* c, *so that*

$$\kappa_A \geq cn \min_{j \in \mathcal{S}} \|M_j\|^2.$$

Therefore, $\mathcal{S}$ *can be almost exactly recovered when*

$$\min_{j \in \mathcal{S}} \|M_j\| \geq c\sqrt{\log p/n}.$$

Part III: Clustering and Regression with the selected covariates

High Dimensional Data Problems

$$(X, A, X_{new})$$

- Key: some users have full network data, and other users only have covariates
- Network data $A \in \mathcal{R}^{n \times n}$
- Covariate data $X \in \mathcal{R}^{n \times p}$ that matches A
- Covariate data $X_{new} \in \mathcal{R}^{(N-n) \times p}$ without network

- Insight: $\hat{S}(A, X_1)$ will improve statistical inference of X_2
- Consider the clustering problem and regression problem

Regression

- Suppose we observe (A, X, z, X_{new})
- Here, $z_i = \alpha^T Y_i + \epsilon_i$. It is on X only
- For X_{new}, we want to estimate $z(X_{new})$

Regression

- Suppose we observe (A, X, z, X_{new})
- Here, $z_i = \alpha^T Y_i + \epsilon_i$. It is on X only
- For X_{new}, we want to estimate $z(X_{new})$

Theorem (Consistency of Regression)

Under RDPG and further condition that $rank(M) = K$, $\lambda_K(M) \geq c\|M\|$ *and* $\kappa_M = \min_{j \in \mathcal{S}} \|M_j\| > 3\frac{\sqrt{n}+\sqrt{\hat{s}}}{\sqrt{ns}}$, *there is*

$$|\hat{\gamma}^T X_{n+1}^{\hat{S}} - \alpha^T Y_{n+1}| \leq \frac{\sqrt{n}+\sqrt{\hat{s}}}{\kappa_M \sqrt{ns}} + C\sigma_\epsilon(\frac{1}{\sqrt{n}} + \frac{1}{\kappa_M \sqrt{s}})$$

Clustering

- Suppose we observe (A, X, X_{new})
- The network info for X_{new} is not available
- Goal: estimate the community labels

Clustering

- Suppose we observe (A, X, X_{new})
- The network info for X_{new} is not available
- Goal: estimate the community labels

Theorem (Consistency of Clustering)

Under DCSBM and regular conditions on the distance between rows of M, then there is

$$Err = \frac{misclassified}{N} \leq \frac{\hat{s} + N}{2Ns\kappa_M^2}$$

Part IV: Numerical Results

DCSBM

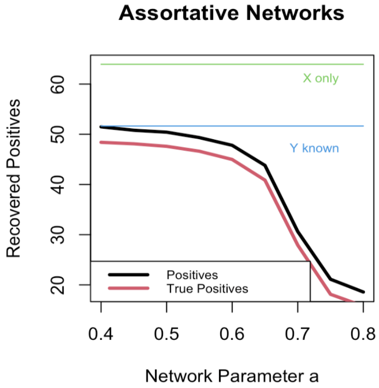
Simulation settings:

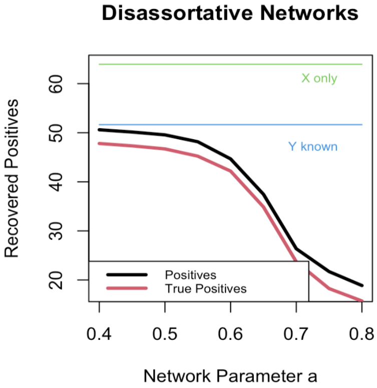
- For $n_1 = 600$ samples, we have A and X_1
- For $n_2 = 400$ samples, we only have X_2
- $p = 1200$ covariates, among them $s = 50$ contribute to the clustering
- $K = 3$
- Connection matrix:

$$B_{assort} = \begin{pmatrix} 1 & a & a \\ a & 1 & a \\ a & a & 1 \end{pmatrix}, \quad B_{dis} = \begin{pmatrix} a & 1 & 1 \\ 1 & a & 1 \\ 1 & 1 & a \end{pmatrix}$$

- Network sparsity: $\theta_i \sim Unif(0.3, 0.5)$
- Covariate signal strength: $M_j \sim N(0.25, (0.05)^2)$

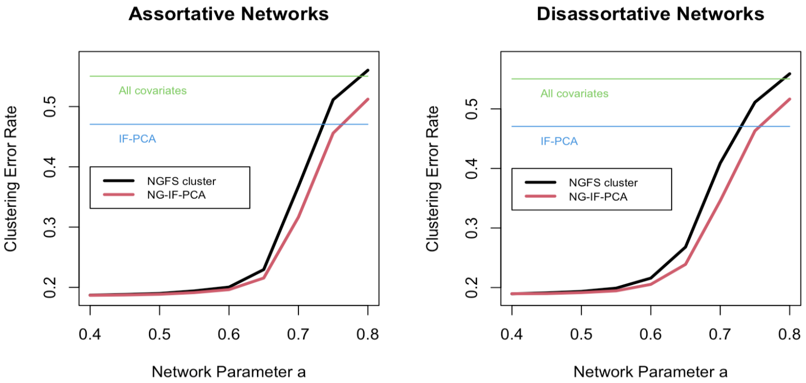
DCSBM: Covariate Selection





- Results for Laplacian is very similar, so I didn't plot it separately

DCSBM: Clustering Errors



- IF-PCA (Jin and Wang (2016)), where a clustering HCT is designed
- NG-IF-PCA: use network-guided p-values, but with the HCT as Jin and Wang (2016)
- Highly depending on the network structure

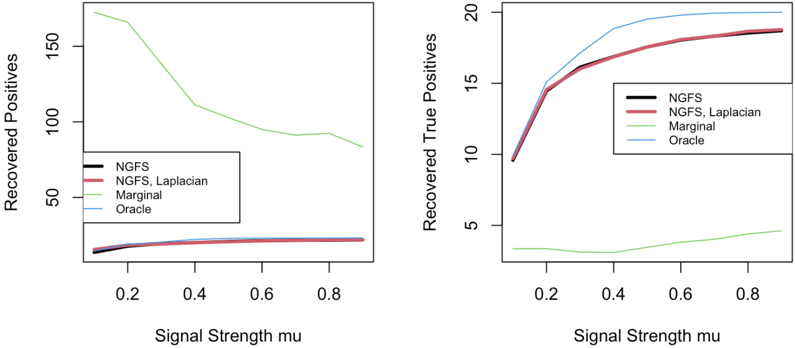
Latent Position Model

Simulation settings:

- For $n_1 = 600$ samples, we have A and X_1
- For $n_2 = 400$ samples, we only have X_2
- $p = 1200$ covariates, among them $s = 20$ contribute to the clustering
- $K = 6$

- $Y_{i,j} \sim Unif(1, 2)$
- $\rho_n = 0.01$ for the network density
- $z = Y\alpha + N(0, 1)$, $\alpha_k \sim Unif(0, 5)$
- $M_{ij} \sim Unif(-\mu, \mu)$ for $j \in \mathcal{S}$

Latent Position Model: Covariate Selection



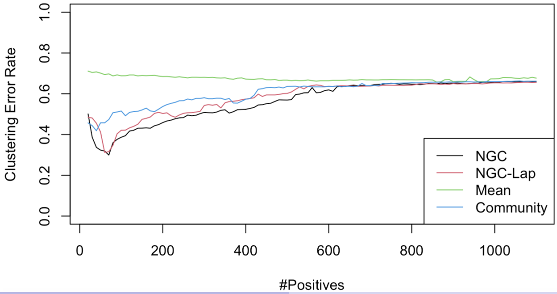
- Marginal *p*-values suffer the large noise in *z* and sparsity

LastFM Data Analysis

- 1839 users with both networks and liked artists
- 760 users with artist list only
- $p = 2201$ artists after we remove those with very few likes
- $K = 5$ countries (we pretend we don't know the truth)
- The covariates are sparse and discrete
  - we use t-test on the group $\xi_k(X_j = 1)$ and $\xi_k(X_j = 0)$ instead
  - The stat. given by X_j only, is taken as $p_j = mean(X_j)$

LastFM Data Analysis: Clustering Error Rate

- Instead of using HCT, we consider a range of $S \in \{20, 30, 40, \cdots, 1100\}$
- We compare the methods
  - Network-Guided Clustering (NGC);
  - Network-Guided Clustering, with Laplacian (NGC-Lap);
  - Select most popular artists and clustering (Mean);
  - Community detection first, and then evaluate covariates by the community detection results (Community)



Discussions

- Nowadays, we observe large amount of data with various formats from one sample, not only the networks and covariates.
    - To combine them, the spectral method worths to try
    - Spectral method is computationally efficient, robust, and even powerful
    - Today we show such an example
- The models of A, X and the latent structure Y should be further explored
    - A: hypergraph, dynamic network
    - X: non-linear relationship between X and Y
    - A and X: partial common latent structure; dependency...

Main paper:

- Network-Guided Covariate Selection and Related Problems Manuscript